CLAIMS

I claim:

1. A collection command applicator process for applying computer commands to one or more collections, comprising the following steps:

   (a) obtaining a list of collections for processing, and

   (b) applying one or more computer commands to collections within said list of collections,

   thereby improving productivity of human workers by enabling them to automatically process large numbers of collections using significantly fewer collection command applicator commands, and thereby providing a solution to the general collection command applicator problem.

2. The process of claim 1, further comprising the step of:

   (a) sorting said list of collections into a visit order before applying computer commands to collections within said list of collections,

   thereby ensuring that commands are applied in proper visit order according to processing interdependencies that may exist among processed collections, and thereby providing a solution to the collection visit order problem.

3. The process of claim 2, wherein

   (a) said step of sorting said list of collections uses information selected from

the group consisting of collection specifier information and collection type definition information and collection content information,

thereby enabling detailed collection information to be used advantageously in visit order calculations, in an automated, scalable way that was not previously possible, and

thereby enabling human knowledge workers to correctly process large sets of collections that require dynamically calculated, complex execution visit orderings that must be based on detailed collection information.

4. The process of claim 1, wherein

(a) said step of obtaining a list of collections uses a collection recognizer means to identify collections for processing,

thereby providing a precise and scalable way to identify interesting collections for automated processing.

5. The process of claim 1, wherein

(a) said step of applying one or more computer commands uses a command execute parallel means to apply commands,

thereby using parallel processing techniques to apply commands to collections in a minimum amount of time, and thereby providing a solution to the parallel collection command execution problem.

6. The process of claim 1, wherein

(a) said step of applying one or more computer commands uses an indirect

command execution means selected from the group consisting of command execution sequential indirect means and command execution parallel indirect means,

thereby creating an efficient, reusable, and persistent way of applying arbitrary commands to a set of collections without incurring collection list production costs for each future command application to the same set of collections.

7. The process of claim 1, wherein

(a) said step of applying one or more computer commands uses nearby execution directory techniques to calculate ultimate command execution directories,

thereby enabling said computer commands to be conveniently applied within nearby execution directories without requiring extra effort from human knowledge workers, and thereby providing a solution to the nearby execution directory problem.

8. A programmable collection command application device for applying computer commands to one or more collections, whose actions are directed by software executing a process comprising the following steps:

(a) obtaining a list of collections for processing, and

(b) applying one or more computer commands to each collection in said list of collections,

thereby improving productivity of human workers by enabling them to automatically process large numbers of collections using significantly fewer

collection command applicator commands, and thereby providing a solution to the general collection command applicator problem.

9. The programmable device of claim 8, further comprising the step of:

(a) sorting said list of collections into a visit order before applying computer commands to collections within said list of collections,

thereby ensuring that commands are applied in proper visit order according to processing interdependencies that may exist among processed collections, and thereby providing a solution to the collection visit order problem.

10. The programmable device of claim 9, wherein

(a) said step of sorting said list of collections uses information selected from the group consisting of collection specifier information and collection type definition information and collection content information,

thereby enabling detailed collection information to be used advantageously in visit order calculations, in an automated, scalable way that was not previously possible, and

thereby enabling human knowledge workers to correctly process large sets of collections that require dynamically calculated, complex execution visit orderings that must be based on detailed collection information.

11. The programmable device of claim 8, wherein

(a) said step of obtaining a list of collections uses a collection recognizer means to identify collections for processing,

thereby providing a very precise, yet scalable way to identify collections for automated processing.

12. The programmable device of claim 8, wherein

(a) said step of applying one or more computer commands uses a command execute parallel means to apply commands,

thereby using parallel processing techniques to apply commands to collections in a minimum amount of time, and thereby providing a solution to the parallel collection command execution problem.

13. The programmable device of claim 8, wherein

(a) said step of applying one or more computer commands uses an indirect command execution means selected from the group of command execution sequential indirect means and command execution parallel indirect means,

thereby creating an efficient, reusable, and persistent way of applying arbitrary commands to a set of collections without incurring collection list production costs for each future command application to the same set of collections.

14. The programmable device of claim 8, wherein

(a) said step of applying one or more computer commands uses nearby execution directory techniques to calculate ultimate command execution directories,

thereby enabling said computer commands to be conveniently applied within nearby execution directories without requiring extra effort from human

knowledge workers, and thereby providing a solution to the nearby execution directory problem.

15. A computer readable memory, encoded with data representing a collection command application computer program that can be used to direct a computer when used by the computer, comprising:

(a) means for obtaining a list of collections for processing, and

(b) means for applying one or more computer commands to each collection in said list of collections,

thereby improving productivity of human workers by enabling them to automatically process large numbers of collections using significantly fewer collection command applicator commands, and thereby providing a solution to the general collection command applicator problem.

16. The computer readable memory of claim 15, further comprising:

(a) means for sorting said list of collections into a visit order before applying computer commands to collections within said list of collections,

thereby ensuring that commands are applied in proper visit order according to processing interdependencies that may exist among processed collections, and thereby providing a solution to the collection visit order problem.

17. The computer readable memory of claim 16, wherein

(a) said means for sorting said list of collections uses information selected from the group of collection specifier information and collection type definition information and collection content information,

thereby enabling detailed collection information to be used advantageously in visit order calculations, in an automated, scalable way that was not previously possible, and thereby enabling human knowledge workers to correctly process large sets of collections that require dynamically calculated, complex execution visit orderings that must be based on detailed collection information.

18. The computer readable memory of claim 15, wherein

(a) said means for obtaining a list of collections uses a collection recognizer means to identify collections for processing,

thereby providing a very precise, yet scalable way to identify collections for automated processing.

19. The computer readable memory of claim 15, wherein

(a) said means for applying one or more computer commands uses a command execute parallel means to apply commands,

thereby using parallel processing techniques to apply commands to collections in a minimum amount of time, and thereby providing a solution to the parallel collection command execution problem.

20. The computer readable memory of claim 15, wherein

(a) said means for applying one or more computer commands uses nearby execution directory techniques to calculate ultimate command execution directories,

thereby enabling said computer commands to be conveniently applied within nearby execution directories without requiring extra effort from human knowledge workers, and thereby providing a solution to the nearby execution directory problem.